

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



A Reliable and Flexible Transmission Method in Wireless Sensor Networks

Dae-Young Kim and Jinsung Cho
Kyung Hee University
S. Korea

1. Introduction

Recent advances in wireless communication have enabled multifunctional tiny nodes to construct a wireless network by themselves Akyildiz et al. (2002). The network is called a wireless sensor network. The tiny sensor nodes are densely deployed in a physical space. They monitor physical phenomena, deliver information, and cooperate with neighbor nodes Akyildiz et al. (2002); Culler et al. (2004); Hac (2003); Zhao and Guibas (2004); Chong and Kumar (2003). The communication systems in end-to-end data transmission of wireless sensor networks employ a recovery mechanism for lost data during data transmissions because reliable data transmissions are required for various sensor network applications.

Two types of retransmission have been proposed for the recovery, namely end-to-end loss recovery (E2E) and hop-by-hop loss recovery (HBH). In these mechanisms, lost packets are retransmitted from a source node or an intermediate node. If a retransmit request for lost packets is sent to a source node, the end-to-end delay may increase because channel error accumulates exponentially over multi-hops Wan et al. (2002). The well-known HBH mechanisms are PSFQ Wan et al. (2002) and RMST Stann & Heidemann (2003). PSFQ is based on ACK message and RMST is on NACK message. In HBH, when intermediate nodes cache data packets into storage, retransmissions can be requested to an intermediate relay node to reduce end-to-end delays. Because sensor nodes have limited resources, however, it is difficult for all sensor nodes to find sufficient space in their routing paths to cache data packets. There is therefore a tradeoff between end-to-end delays and memory requirements.

Because data traffic on sensor networks requires a variety of levels of communication reliability (CR) depending on the application, a loss recovery method to guarantee the desired CR should be provided. Traditional loss recovery mechanisms consider only 100% reliability. In this letter, we propose a flexible loss recovery mechanism to guarantee various CRs and we discuss the tradeoff between end-to-end delays and memory requirements for various CRs. The proposed method can be widely used for the design of wireless sensor networks that require a variety of CRs.

2. A Reliable and Flexible Transmission Method in Wireless Sensor Networks: Active Caching

As mentioned previously, E2E involves large end-to-end delays for 100% reliability because of high packet loss during multi-hop transmissions. To guarantee high reliability and minimal

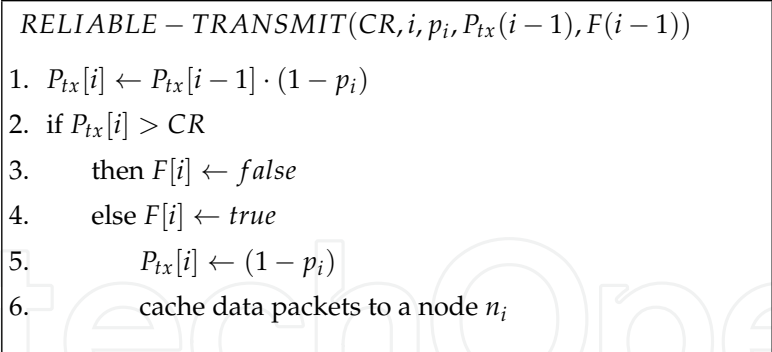


Fig. 1. Active caching algorithm at i -th node, n_i .

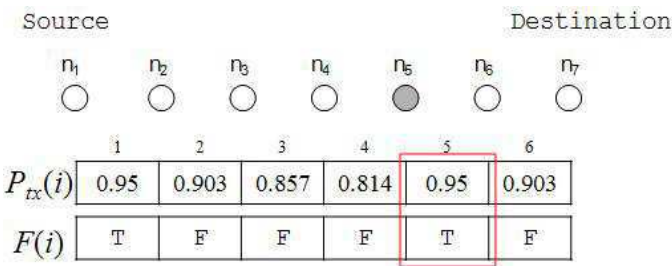


Fig. 2. An example of active caching.

end-to-end delays, HBH caches data in every node over a routing path resulting in large memory requirements. When only some nodes cache data on a routing path, there exists a tradeoff between the end-to-end delays and the memory requirements. For applications which do not require 100% reliability, every node needs not cache data via HBH. When a target CR is given, we need a flexible method to guarantee the given CR while minimizing the memory requirement. In this section, we present such a method - active caching (AC).

The proposed scheme allows various CR s of application services. It determines positions where data caching occurs using a dynamic programming algorithm, which solves every subproblem just once and then saves its answer in a table to avoid the work of recomputing the answer Cormen et al. (2001). If there are holes in sequence numbers of received data, a caching node recognizes packet loss Karl & Willig (2005). The caching node sends a NACK message to a previous caching node along the path and the previous caching node retransmits lost packets selectively.

First, we define the problem and subproblems for the active caching as a dynamic programming algorithm to guarantee an end-to-end reliable data transmission as:

Problem: $P_{tx}(H) > CR$.

Subproblem: $P_{tx}(h) > CR$, where $h = 1, 2, \dots, H$.

The packet delivery rate $P_{tx}(H)$ during total hop counts H should be greater than the desired communication reliability CR . To do that, the packet delivery rate $P_{tx}(h)$ during hop counts h in each hop should be greater than the CR . The key idea for solving the problem is to cache data packets if the probability of packet transmission does not satisfy the desired communication reliability. By solving the subproblems, we can solve the entire problem.

Figure 1 shows the proposed active caching algorithm for loss recovery. Each node solves the subproblem using the tables for the packet delivery rate $P_{tx}(i)$ until i -th hop and the caching flag of i -th node $F(i)$. Both $P_{tx}(i-1)$ and $F(i-1)$ of the tables are piggybacked in data packets and they are delivered to the next node. In a source node ($i = 1$), $P_{tx}(1)$ is $1 - p_1$ as the packet delivery rate at the 1st hop and $F(1)$ is *true*. *Line 1-3*: n_i calculates $P_{tx}(i)$ using $P_{tx}(i-1)$, where $P_{tx}(i)$ accumulates the packet delivery rate $1 - p_i$ of i -th hop while packets are transmitted. After that, it compares $P_{tx}(i)$ with CR . If $P_{tx}(i)$ satisfies the desired CR , n_i is not a caching node ($F(i)$ is *false*). *Line 4-6*: If $P_{tx}(i)$ does not guarantee the desired CR , n_i becomes a caching node ($F(i)$ is *true*). In this case, $P_{tx}(i)$ compensates for its packet delivery rate as the reliability instead of accumulating $P_{tx}(i)$ and data packets are cached onto n_i 's buffer. Each node runs the algorithm of Figure 1 and the total active caching over a routing path is performed by the dynamic programming algorithm. Figure 2 shows an example of the active caching when seven sensor nodes are deployed sequentially and they have an average 5% packet loss rate and 80% CR . Every node satisfies 80% CR and data caching occurs at n_5 . When packet loss happens between a source node n_1 and the caching node n_5 , the caching node requests retransmission to the source node. When packet loss happens between the caching node and a destination node n_7 , the destination node requests retransmission to the caching node.

3. Analysis

A packet loss rate occurs due to wireless link and contention errors. Since all the packets are destined to the sink node in wireless sensor networks, the contention error in links close to the sink node may increase. To model the packet loss rate at i -th hop, we assume the uniform link error p_l and the contention error which is proportional to the square of transmission hop counts.

$$p_i = p_l + \alpha i^2, \quad (1)$$

where α is the contention failure factor. Then the packet delivery rate during h hops from the s -th node is

$$P_{tx}(s, h) = \prod_{i=s}^{s+h-1} (1 - p_i). \quad (2)$$

Data caching occurs when $P_{tx}(s, h)$ is lower than CR . When the number of nodes N over a route and CR are given, the hop counts h from a caching node s and the number of caching nodes N_c are obtained by the function in Figure 3. Φ represents a set of (s, h) tuples and the (s, h) tuples are used to compute the retransmission counts of lost packets. For example in Figure 2, $\Phi = \{(1, 4), (5, 2)\}$.

$$\Phi = \{(s_j, h_j) \mid j = 1, \dots, N_c\}. \quad (3)$$

If the retransmission counts for h hops from a caching node s is given by $\psi(s, h)$, the total retransmission counts $E[C]$ between a source node and a sink node are represented by the sum of $\psi(s, h)$ as

$$E[C] = \sum_{j=1}^{N_c} \psi(s_j, h_j). \quad (4)$$

Because the retransmitted packets can also experience transmission failure, we should consider repeated retransmissions for $\psi(s, h)$. Let $\Gamma_f(j, s, h)$ indicate the number of transmitted packets at the j -th retransmission. Then $\psi(s, h)$ can be represented as

```

CalcHopCounts(N, CR)
1.  $n \leftarrow 1, s \leftarrow 1, h \leftarrow 1, N_c \leftarrow 0$ 
2.  $\Phi = \phi$ 
3. loop:  $n < N$ 
4.   if  $P_{tx}(s, h) > CR$ 
5.     then  $n \leftarrow n + 1, h \leftarrow h + 1$  //no caching
6.   else  $h \leftarrow h - 1$  //caching
7.     if  $(h = 0)$ 
8.       then  $h \leftarrow 1, n \leftarrow n + 1$ 
9.     add  $(s, h)$  to  $\Phi, N_c \leftarrow N_c + 1$ 
10.     $s \leftarrow n, h \leftarrow 1$ 
11. end loop
12. if  $(h > 1)$ 
13.   then add  $(s, h - 1)$  to  $\Phi, N_c \leftarrow N_c + 1$ 

```

Fig. 3. Function to obtain (s, h) tuples.

$$\psi(s, h) = \sum_{j=1}^{\infty} (h \cdot \Gamma_f(j, s, h) \cdot P_{tx}(s, h)). \quad (5)$$

If we let $\Gamma_s(k, s, h)$ be the number of successfully transmitted packets among k packets during h hops from node s , $\Gamma_f(j, s, h)$ can be represented recursively as

$$\Gamma_f(j, s, h) = \Gamma_f(j - 1, s, h) - [\Gamma_s(\Gamma_f(j - 1, s, h), s, h)]^1, \quad (6)$$

where $\Gamma_f(0, s, h) = K$ and K is the number of total packets which is generated in a source node.

The number of successfully transmitted packets $\Gamma_s(k, s, h)$ can be calculated by the probability of successful transmission of Bernoulli trials $P_s(k, m, s, h)$ as

$$\Gamma_s(k, s, h) = \sum_{m=1}^k m \cdot P_s(k, m, s, h). \quad (7)$$

If m data packets are transmitted successfully among k packets to deliver across h hops from a caching node s , the probability of successful transmissions can be obtained by Bernoulli trials as

$$P_s(k, m, s, h) = \binom{k}{m} \cdot P_{tx}(s, h)^m \cdot (1 - P_{tx}(s, h))^{k-m}. \quad (8)$$

The memory requirement B is defined as the caching rates of intermediate nodes including a source node. It is computed by N_c and the number of relay nodes over a routing path:

$$E[B] = \frac{N_c}{N - 1}. \quad (9)$$

¹ $[x]$ is n , in case of $n - 0.5 \leq x < n + 0.5$

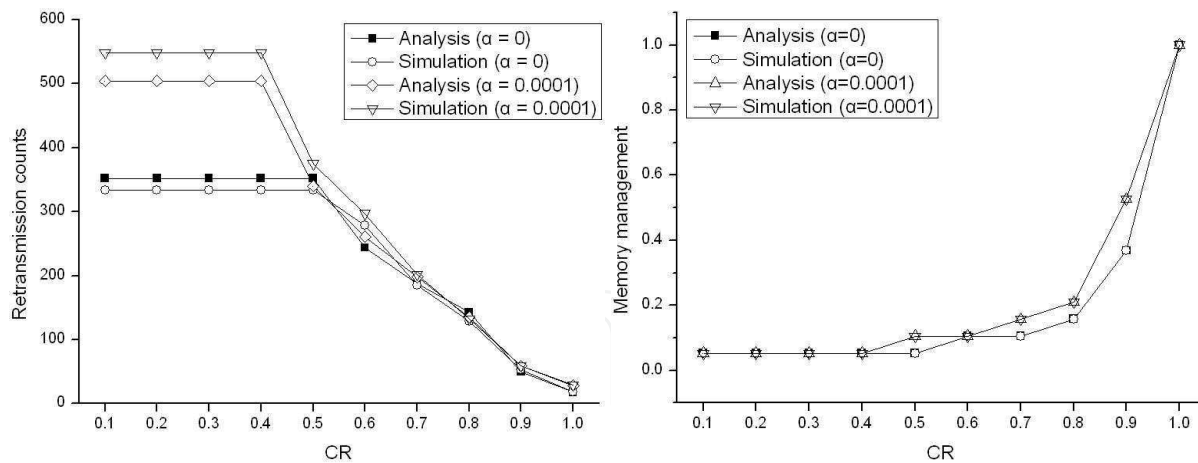


Fig. 4. Validation of our analysis ($p=0.03$).

A high $E[C]$ indicates large end-to-end transmission delays and $E[B]$ represents the memory requirements of buffers on the data transmission routes. Because both $E[C]$ and $E[B]$ can be estimated by CR of traffic through Eq.(4) and Eq.(9), a flexible data transmission system can be designed.

4. Evaluation

In this section, we validate the analysis through simulations and compare the performance of active caching (AC) with that of E2E and HBH. For the simulation, we assume 20 sensor nodes are deployed sequentially and the wireless channel has both link and contention error as described in Section 3. The contention failure factor α is determined as 0.0001 by considering total hop counts. So, p_i in Eq.(1) ranges from 0.03 to 0.07 when p is 0.03 in our experiments. The sensor nodes employ AODV as a routing protocol. Assuming a packet is 30 bytes and the data rate is 250kbps, we perform the analysis and simulation by varying CR from 10% to 100%. AC with CR from 0.1 to 1 is expressed as AC0.1 to AC1.

Figure 4 shows the results of the analysis and the simulation of the retransmission counts and the memory requirements when a source transmits 40 packets. The results of the analysis and the simulation show an average of 94% similarity. Figure 4 also represents the tradeoff as mentioned earlier. The high CR requires a high memory requirement for reliability and it decreases the retransmission counts. When the memory requirement is the lowest, the retransmission counts are the highest and AC runs as E2E. In short, we can design wireless sensor networks that take the desired CR and memory requirements into consideration through the proposed active caching.

Figure 5 shows the performance comparison of E2E, HBH, and AC. Because AC with the highest memory requirement caches data to every intermediate node, it operates as HBH. When AC does not perform data caching, it operates as E2E. That is, AC switches between HBH and E2E while showing the performance tradeoff between them. In addition, it has a tolerable end-to-end delay to minimize the memory requirement depending on CR. In Figure 5, the end-to-end delays of E2E increase when the wireless channel has a high link error rate. However, the end-to-end delay of AC maintains similar values because AC increases the memory requirements to ensure CR. An evaluation has been performed for 10 and 50 nodes

deployed over a route, and the results are similar to the case of 20 nodes. These results have been omitted due to the page limitation.

Figure 6 shows the ratio of caching nodes over relay nodes. Because the contention error increases when the density of nodes increases, the ratio of caching nodes increases when the number of sensor nodes increases.

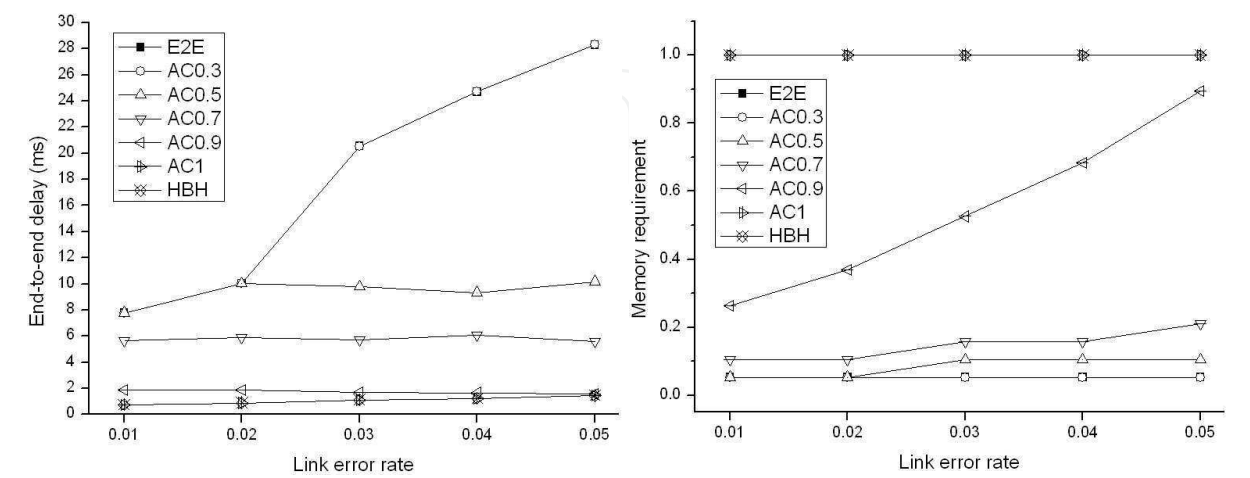


Fig. 5. Performance comparison of E2E, HBH, and AC.

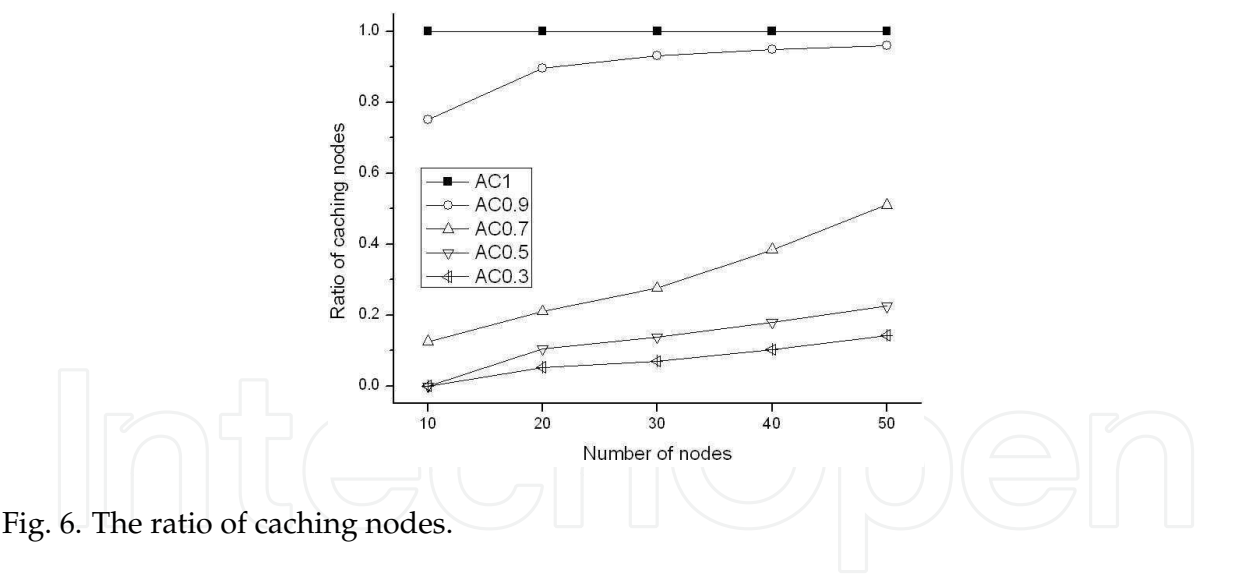


Fig. 6. The ratio of caching nodes.

5. Conclusion

Wireless sensor networks transmit data through multiple hops. End-to-end data transmission must recover lost data for reliable data transmissions. Active caching (AC) provides more flexible end-to-end delays and memory requirements for a given reliability than the existing recovery mechanisms (i.e., E2E, HBH). By using the proposed dynamic loss recovery with active caching, a flexible end-to-end data transmission system can be designed.

6. Acknowledgement

This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency)" (NIPA-2010-(C1090-1021-0003))

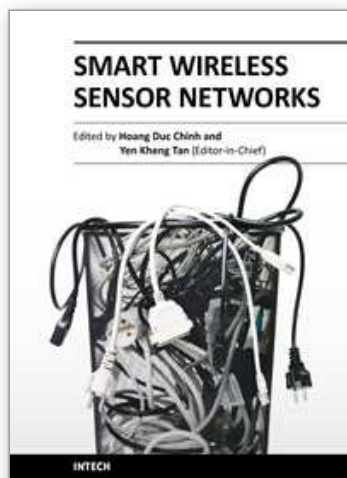
7. References

- Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). A survey on sensor networks, *IEEE Communications Magazine*, Vol. 40(No. 8): pp. 102–114, August 2002.
- Culler, D., Estrin, D., and Srivastava, M. (2004). Guest editors' introduction: Overview of sensor networks. *IEEE Computer*, Vol. 37(No. 8): pp. 41–49, August 2004.
- Hac, A. (2003). *Wireless sensor network designs*, John Wiley & Sons, 2003.
- Zhao, F. and Guibas, L. (2004). *Wireless sensor networks: An information processing approach*, Morgan Kaufmann Publishers, 2004.
- Chong, C. -Y. and Kumar, S. (2003). Sensor networks: Evolution, opportunities, and challenges, *Proceedings of the IEEE*, Vol. 91(No. 8): pp. 1247-1256, August 2003.
- Wan, C. Y., Campbell, A. T., and Krishnamurthy, L. (2002). PSFQ: A reliable transport protocol for wireless sensor networks, *Proceedings of ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 1-11, September 2002.
- Stann, F. and Heidemann, J. (2003). RMST: Reliable data transport in sensor networks, *Proceedings of IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 102-112, May 2003.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms*, Vol. 1, The MIT Press, 2001.
- Karl, H. and Willig, A. (2005). *Protocols and architectures for wireless sensor networks*, John Wiley & Sons, 2005.

IntechOpen

IntechOpen

IntechOpen



Smart Wireless Sensor Networks

Edited by Yen Kheng Tan

ISBN 978-953-307-261-6

Hard cover, 418 pages

Publisher InTech

Published online 14, December, 2010

Published in print edition December, 2010

The recent development of communication and sensor technology results in the growth of a new attractive and challenging area – wireless sensor networks (WSNs). A wireless sensor network which consists of a large number of sensor nodes is deployed in environmental fields to serve various applications. Facilitated with the ability of wireless communication and intelligent computation, these nodes become smart sensors which do not only perceive ambient physical parameters but also be able to process information, cooperate with each other and self-organize into the network. These new features assist the sensor nodes as well as the network to operate more efficiently in terms of both data acquisition and energy consumption. Special purposes of the applications require design and operation of WSNs different from conventional networks such as the internet. The network design must take into account of the objectives of specific applications. The nature of deployed environment must be considered. The limited of sensor nodes’ resources such as memory, computational ability, communication bandwidth and energy source are the challenges in network design. A smart wireless sensor network must be able to deal with these constraints as well as to guarantee the connectivity, coverage, reliability and security of network’s operation for a maximized lifetime. This book discusses various aspects of designing such smart wireless sensor networks. Main topics includes: design methodologies, network protocols and algorithms, quality of service management, coverage optimization, time synchronization and security techniques for sensor networks.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Jinsung Cho and Dae-young Kim (2010). A Reliable and Flexible Transmission Method in Wireless Sensor Networks, Smart Wireless Sensor Networks, Yen Kheng Tan (Ed.), ISBN: 978-953-307-261-6, InTech, Available from: <http://www.intechopen.com/books/smart-wireless-sensor-networks/a-reliable-and-flexible-transmission-method-in-wireless-sensor-networks>

INTech
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820

www.intechopen.com

Fax: +385 (51) 686 166
www.intechopen.com

Fax: +86-21-62489821

IntechOpen

IntechOpen

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen